BOLETIN CUQ NUMERO 4 ENERO 1989.

CONTENIDO

INFORMACION SOBRE CUQ

Cada colaborador debe enviar un disco 3.5" con alguna colaboración junto a un sobre con los sellos de retorno pegados y sus señas de retorno escritas (Destinatario y Remite).

Las colaboraciones deben ser enviadas a:

Salvador Merino
Ctra. Cádiz, Cerámicas Mary
29640 Torreblanca del Sol
Fuengirola (Málaga)
Teléfono 475043

¿ Cuál es el formato de las colaboraciones ?

Son ficheros QUILL con el margen izquierdo a 0, el sangrado a 5, y el derecho a 80. Margen superior e inferior a 0. 66 lineas por página. No pie de página (OPCIONAL). Solamente usar caracteres normales (No negrita, subrayado,..... Pero podria ser opcional).

¿ Qué se puede enviar ?.

Muy simple cualquier cosa (Cartas, preguntas, programas, trucos, comentarios, criticas, anuncios, articulos.... en otras palabras, DE TODO, y sin censura).

El último Domingo de cada mes se grabaran los discos con el último boletín CUQ y programas donados.

EDITORIAL
=========

Es increible, este es el número cuatro y yo con estos pelos. No me lo puedo creer.

El último Martes Trece murió envenenado mi único Gato. Aunque no soy superticioso, creo que podria darnos bastante suerte.

Creo que me he perdido, pero os recomiendo que leais las noticias de Navidad.

Como es natural, voy a dar una respuesta a la opinión de Marcos Cruz acerca de CUQ.
Por lo pronto tiene razón en todo (no se le ha olvidado de ningún detalle).
Si alguien está interesado en colaborar, y lo demuestra, siempre encontrara un hueco libre aunque seamos más de 20. Pero quienes no deseen aportar nada, simplemente beneficiarse sin aportar nada a cambio, serán eliminados sin piedad para que otro ocupe su puesto (con alguna excepción por supuesto).
Tal como yo veo las cosas, el único remedio, para que CUQ llegue al mayor número de usuarios, es que los usuarios que consigan una copia hagan copias a sus amigos más cercanos ya sean en MDV o PAPEL.
Otro remedio para que se beneficie más gente de CUQ seria ofrecer el boletín via MODEM. Los únicos problemas serian la factura del teléfono para el interesado (incluido tener teléfono), tener MODEM (yo no tengo aún, pero no seria un problema grave. El verdadero problema es que tengo mi teléfono particular a 2 KM de mi QL, y el de mi padre a unos 80 M.), cualquier usuario de otro ordenador podria colarse,....
Lo único que me queda por decir es que se admiten nuevas ideas.

Si seguimos al mismo ritmo que actualmente (esto parece una carrera de GALGOS), no me extrañaria nada que en el futuro el material de un mes ocupe o rebase la capacidad de un disco 720 K. Con este boletín 4 hemos llenado el primer disco 3.5" 720 K.

S. Merino

NOTICIAS DE NAVIDAD

Digital Precision ha anunciado su emulador MS-DOS y parece que va a emular también parcialmente alguna tarjeta gráfica, pero habrá que esperar a enero.

Otra casa, Ant Computing, ha anunciado la venta de su emulador PC. Solamente emula el modo texto 80 columnas. Está escrito en Assembler. Lee/escribe/formatea en formato IBM. Se podria decir que emula al 100% un IBM PC original de cuando nació el QL.

Hay rumores de que la casa Schon va a lanzar también su emulador MS-DOS, seguramente de Ultrasoft (Alemania).
Como ya podreis observar, esto parece la fiebre del MS-DOS. Somos los últimos, pero vamos a tener más emuladores de MS-DOS que otros 68K (también hay que agregar la chapuza del emulador de SPEM).

Pero la gran noticia, aunque ya sabia algo, es que CST se ha mudado a Dinamarca y ha puesto alli el THOR XVI otra vez en producción. Y ha vendido los derechos de producción y venta de sus viejos productos QL a ABC por un año con derecho a renovación.
Entre los nuevos productos ABC para el QL, hay una nueva tarjeta que incorpora interface disco e interface disco duro, toolkit II y Ataridos (un programa que formatea/lee/escribe discos en formato IBM y Atari ST).

La mayor garantia de futuro para un usuario de QL es que el THOR continue fabricandose. Solamente en los últimos meses ya ha aumentado considerablemente el número de titulos nuevos.

Por lo visto Diego Alcalá y Lorenzo Ayuda han conseguido imprimir en un MAC los dos boletines CUQ que envie. Y por lo que me han dicho gente que creen haber visto el original , el boletín tiene letra de MUJER y es tamaño folio. Y Diego pide que renovemos la cuota. ¡ Tendrá cara el tio !. 4000 ptas por 3 boletines y un cuarto que no sé si veré algún dia.

La idea de CUQ ha tenido demasiado éxito. La gente quiere una copia de esos boletines y programas. Pero colaborar es algo que habria que ponerlo en duda (si no colaboran, aunque sea escribiendo una pregunta, no veran el boletín). E incluso me han llegado rumores de que un Club de Atari ST está interesado por CUQ, pues algunos de sus socios han comprado recientemente el QL Emulator (no me extrañaria nada que SPEM vendiese copias de CUQ en Italia).

Según Alan Sugar, ha precisado que se adheriría al acuerdo de los constructores sobre el bus EISA, que tiende a contrapesar el busMCA de los PS/2 de IBM. Pero se alineará en el estándar vencedor, con riesgo de royalties a IBM. Sin confusión, sin riesgo, sin sorpresa. El constructor inglés construye con hormigón.

Este último texto ha sido sacado de Ordenador Personal, revista que se ha convertido en una mas exclusiva al mundo PC (antes era neutral y para todos los usuarios, pero asi van las cosas). Pero si esa es la estrategia de Mr. Sugar, hay que recordar que aún tiene los derechos del Sinclair QL que supongo que no dudaria en usarlos para construir nuevas versiones del QL si el THOR y demás sistemas compatibles QDOS consiguen el éxito que se va haciendo grano a grano. Un QL basado totalmente en el MC68020 o 68030 (igual que el MAC II) podria ser la solución a todos esos bus de 32 bits de la competencia PC (ahora nosotros les llevamos 5 años de ventaja).

Salvador Merino
Fuengirola (Málaga)
Diciembre 1988.

PROYECTO FANZINE

Anuncio el proyecto de edición de un fanzine sobre "informática general especulativa", con atención preferente a: familia 680XX, Forth, Inteligencia Artificial, modems, nuevas tendencias en hard, y todo lo que se nos ocurra; y con "desantención" preferente a: Amstrad, PCs, y demás calaña.

De momento es sólo un proyecto que estamos configurando por aquí un par de colegas. Cuando esté en marcha os lo comunicaré y recibiréis ejemplares.

En principio estamos pendientes de la adquisición del nuevo "Professional Publisher" de Digital Precision para meternos en faena, con el objeto de hacer las cosas en condiciones. Disponemos de un QL con Trump Card y un Amiga 500.

El objetivo es aportar material original desde el principio, aunque en último caso recurriríamos -para el primer número de prueba- a incluir algo interesante que haya aparecido en CUQ. Sea como sea, os mantendré al corriente de todo.

LA ETERNA PEDANTERIA DE IBM

En estas fechas estamos -y vamos a seguir estando- torturados por un pretencioso anuncio televisivo de IBM proclamando a bombo y platillo su engreido Personal System 2 con su más engreido aún OS/2.

El fracaso de los PS/2 y del OS/2 se está cuajando inevitablemente. Se ha tirado demasiado de la familia Intel para mantener la compatibilidad dentro de unos límites, cosa imposible si se quieren hacer innovaciones auténticas.

En primer lugar, ante la pedantería de IBM al proclamar su ostentoso sistema multitarea, hay que dejar clara constancia de que el primer ordenador que pudo hacer eso -a nivel de informática personal y doméstica- fue el Sinclair QL. Ahora nos quieren presentar como novedad universal algo que el QL hace desde su nacimiento 4 o 5 años atrás, y además mucho mejor.

En segundo lugar, ahí van las gordísmas pegas de los nuevos modelos PS/2 y su "novedoso" sistema operativo OS/2:

-El OS/2 no es multitarea multiusuario como se anunció en principio, sino sólo multitarea monousuario.

-El OS/2 no puede ejecutarse sobre PC XT con tarjeta acelaradora. No obstante, se anuncian para dentro de un tiempo tarjetas aceleradoras para XT que permitan ejecutar el OS/2

-El OS/2 puede correr sobre un PC AT con unidad de disco de alta capacidad (1,2 Mbytes en 5,25 o 1,44 Mbytes en 3,5), un disco duro con 4 o 5 Mbytes libres, y 1,5 Mbytes de RAM... ¡Pero para ejecutar desde el OS/2 aplicaciones del MS-DOS hay que añadir los 640 Kbytes propios del mundo real del MS-DOS! En suma, para poder correr aplicaciones MS-DOS bajo OS/2 se recomienda tener entre 2,5 y 3 Mbytes de RAM.

-¿Cómo es que el OS/2 no precisa un modelo PS/2 para ejecutarse? Respuesta: porque el OS/2 no hace ningún uso de la nueva arquitectura Micro Channel de los Personal System 2.

-Intentar ejecutar el OS/2 en un AT que no sea IBM puede ser poco menos que imposible. Por ello, las mayores marcas de compatibles IBM PC tienen que comprar los derechos del OS/2 a Microsoft para producir sus propias versiones.

-El OS/2 puede correr en ordenadores provistos del 80386 en la medida en que el 80386 puede emular al 80286, es decir, el OS/2 no aprovecha ninguna de las ventajas del 80386 respecto del 80286.

-¿Cuántas aplicaciones pueden correr simultáneamente bajo OS/2? Respuesta: una famélica docena. ¡Qué risa le está entrando a mi QL!

En resumen, habremos de sufrir durante años el dominio IBM-Microsoft-Intel, pero yo apuesto por la familia Motorola 680XX y por su total desarrollo en los próximos tiempos, hasta que se implante el proceso en paralelo, los transputers y demás.

Cuando la normalidad sean los ordenadores con proceso en paralelo, con una memoria enorme, discos ópticos y EPROMs como sistemas de almacenamiento auxiliar, etcétera, IBM reinventará la rueda de nuevo (como ha hecho con la adopción de los discos de 3,5 y con la multitarea) y nos la presentará como novedad.

COLECTIVO DE PERIODISMO CIENTIFICO APEIRON

El colectivo Apeiron edita un fanzine de igual nombre, sobre temas científicos, informática incluida. También mantiene un servicio de documentación, con manuales en castellano de diversos sistemas operativos y programas; y direcciones de centros de investigación de todo el mundo.

La suscripción al fanzine por tres números cuesta 10 sellos de 20 pesetas.

Para suscribirse o cualquier otro asunto, la dirección es:

C.T.E.G.
Ref.: APEIRON
APARTADO 4111
14080 CORDOBA

Marcos Cruz, (QLave-242)
Madrid, Diciembre 1988

ACERCA DE C.U.Q.

Llevamos ya tres boletines elaborados entre todos con esta historia y quisiera hacer algunas puntualizaciones acerca del sistema en sí, para ver si pensamos en algunas mejoras a nivel técnico u organizativo, o símplemente se nos ocurren nuevas formas de hacer algunas cosas.

Primero, pegas que observo:

1. La más grave: que el sistema de distribución "restrictiva" es un poco defraudante. La supuesta "arborescencia" de la distribución no creo que se dé en la práctica. El hecho de que el boletín sólo alcance a una minoría mínima y minoritaria es algo para lo que habría que pensar soluciones.

2. Marginaciones a priori que se producen. Dicho de otro modo, usuarios de QL que, aunque quisieran, no podrían colaborar ni recibir el boletín:

-Quien no tiene unidad de discos.
-Quien no tiene ampliación de memoria (por el tamaño del fichero).
-Quien no tiene impresora (no necesariamente, pero casi).

Segundo, ventajas:

1. El coste por recibir el boletín es mínimo: 26 + 26 pesetas en sellos y un sobre. Esto, siempre que el colaborador haga un solo envío con sus aportaciones y reciba a cambio el nuevo boletín definitivo.

2. El disco permite la inclusión de programas o dibujos realizados por los colaboradores, lo cual es un avance enorme con respecto a ciertas librerías de triste recuerdo...

Tercero, propuestas y posibles soluciones:

1. Acerca de las "marginaciones" hay poco que hacer. Igual que Salvador recibe las colaboraciones y envía los boletines en disco, alguien podría hacerse cargo de hacer lo propio -una vez en sus manos el boletín- pero en microdrives, y alguien más en papel. Así, quienes no tengan impresora, mandarían sus colaboraciones y "solicitudes" de boletín al "sub-distribuidor de la versión papel"; quienes no tuviesen unidad de discos harían lo propio con el "sub-distribuidor de la versión microdrive". Me temo que es un poco tinglado y que ninguo querremos complicarnos la existencia.

2. Sobre el precio, sí hay dos cosas que podrían hacerse. En primer lugar, habría que aprovechar al máximo el espacio del disco, pues por mucho que se llene no pesa más (mientras que si el boletín fuese impreso sí pasaría eso). Esto quiere decir que, siempre que haya material, no debe limitarse la capacidad del fichero principal de texto (pero para eso harían falta muchas colaboraciones y por ende muchos más miembros del invento). Además, habría que organizar las fechas de salida y recepción de envíos (atento, Salvador) para que cada colaborador sólo tuviese que hacer un envío. Esto creo que sería fácil de organizar, y evitaría el desperdiciar tiempo y sellos a todos (Salvador incluido). Por ejemplo: Salvador podría recibir las colaboraciones (un único disco al mes de cada colaborador) a lo largo de un mes. Así, pongamos por caso, el último fin de semana de cada mes Salvador cogería todos los discos recibidos hasta entonces, uniría los textos para crear el boletín, y los echaría al correo. De este modo, todos recibiríamos a primeros del mes siguiente el nuevo boletín correspondiente a dicho mes, en el disco que enviamos "repletito" el mes anterior. Creo que es una estupenda idea. Salvador: piénsalo y organízalo como mejor te convenga -creo que el ejemplo que he contado sería factible- y haz una editorial explicando el asunto para que nos enteremos y nos organizemos.

Por otra parte, no creo que exista ningún inconveniente en abrir el CUQ a quien quiera entrar. Sólo dos normas, a mi parecer:

1. El que no envía un disco con alguna colaboración -por pequeña que sea- no recibe el boletín de ese mes. Eso tiene que estar clarísimo, pues el motivo de todo esto es el INTERCAMBIO de información entre usuarios.

2. Hay que enviar obligatoriamente, junto con el disco de las colaboraciones, un sobre con las señas y los sellos puestos para el retorno.

Si se llevan tales normas a la práctica, creo que el trabajo de unir las colaboraciones y grabar el fichero en los discos y echarlos al correo no aumenta mucho porque aumente el número de colaboradores. ¿No te parece, Salvador?

Por último, ruego a los colaboradores que lean lo que escriben ante de enviarlo, al objeto de minimizar los fallos de redacción y puntuación (¡y de tildes!).

Marcos Cruz, (QLave-242)
Madrid, Diciembre 1988

DESCASTELLANIZAR EL QUILL

El comando TRA sirve para alterar algunos códigos en las salidas serie del QL. TRA 1 activa esta posibilidad y TRA 0 la desactiva. La misión de las alteraciones es cambiar los caracteres españoles del juego de caracteres del QL por los caracteres españoles estandar del código ASCII, para que salgan correctamente en la impresora.

Veamos: existen ciertos caracteres del código ASCII que, dada cierta orden a la impresora (es decir, enviados ciertos caracteres de control) son sustituidos por ciertos caracteres españoles.

Código ASCII	Carácter estandar	Carácter español
91	(corchete abierto)	¡
92	(barra inclinada)	Ñ
93	(corchete cerrado)	¿
123	(llave abierta)	(diéresis)
124	(barra vertical)	ñ

Los caracteres de control que producen ese cambio en la impresora varían según ésta, pero es probable que sean: 27, 55, 7, es decir, que habría que hacer desde Basic:

```
OPEN#3,SER1:PRINT#3,CHR$(27);"7";CHR$(7);
```

O bien, si tenemos el Toolkit II:

```
OPEN#3,SER1:BPUT#3,27,55,7
```

No obstante, la selección del juego de caracteres puede seleccionarse por medio de algunos microinterruptores en la impresora, con lo cual nada más encenderla ya estará en disposición de imprimir los caracteres españoles en lugar de los estandar.

Bueno, ¿y qué hace TRA entonces?

En el juego de caracteres del QL, los caracteres españoles están "por ahí" desperdigados entre los demás, de modo que si, suponiendo que no hemos hecho TRA 1, imprimimos alguno de ellos...

```
PRINT#3,"Ññ¡¿"
```

...lo que pasará es que la impresora imprimirá caracteres gráficos propios, y lo mismo para la cedilla, la "u" con diéresis y las vocales acentuadas.

TRA 1, entonces, lo que hace es traducir ciertos caracteres del QL para que se impriman bien en la impresora:

Carácter enviado	Se traduce a
Ñ	(barra inclinada)
ñ	(barra vertical)
¡	(corchete abierto)
¿	(corchete cerrado)
ü	u + BSP + (diéresis)
Ü	U + BSP + (diéresis)
á	a + BSP + '
é	e + BSP + '
í	i + BSP + '
ó	o + BSP + '
ú	u + BSP +
(corchete abierto)	(
(corchete cerrado)	)

...y alguno más, supongo, como las llaves.

(BSP es la notación de un carácter ASCII 8 que produce un retroceso de un caracter en la cabeza de impresión)

Así, estando la impresora con el juego de caracteres españoles seleccionado y ordenado un TRA 1, todo va a las mil maravillas. Eso es lo que hace Quill.

La pega está en si queremos disponer de los corchetes cuadrados (en Forth se usan bastante) o de las llaves sin perder la posibilidad de imprimir los caracteres españoles más importantes. ¡Entonces habrá que descastellanizar Quill!

Lo primero que hay que hacer es eliminar la orden TRA 1 del fichero boot del Quill.

Lo segundo que hay que hacer es ejecutar el programa install_bas que viene con Quill. Es de todos conocido, de modo que sólo explicaré por encima lo que hay que hacer, en la configuración de la impresora seleccionada:

Como orden preámbulo, incluir los caracteres de control que seleccionen el juego de caracteres inglés o estadounidense -mejor el de USA, que tiene "#" en lugar del signo de la libra esterlina- en la impresora. De este modo, dispondremos del juego ASCII normal, sin que los corchetes cuadrados ni las llaves se nos conviertan en paréntesis.

Ahora, el Quill reserva espacio en cada definición de impresora para traducir hasta 10 caracteres a nuestra elección. En ellos hay que meter los diez caracteres españoles que más se usen: á, é, í, ó, ú, Ñ, ñ, ¡, ¿, ü... ¡no hay espacio para más!

Hay que incluirlos seguido de su traducción, es decir, de los caracteres que ordenen a la impresora que los imprima. En el caso de las vocales acentuadas o la "ü", hay que traducirlas a la vocal, un retorno de carro y la tilde o la diéresis (en este último caso, habrá que incluir la selección del juego de caracteres españoles, a menos que haya una diéresis en el juego de caracteres gráficos de la impresora, y luego volver a seleccionar el juego de caracteres anglosajón). En el caso de los caracteres de puntuación y las "eñes", habrá que incluir los códigos de selección del juego de caracteres español, el carácter correspondiente en ese juego, y los códigos de selección del juego de caracteres anglosajón... La pena es que sólo haya espacio para diez traducciones, de modo que no se puede incluir la cedilla ni la "Ü", ni obtener el signo de la libra o "#", según esté seleccionado el juego de caracteres USA o England,

respectivamente.

Bueno, la verdad es que todo esto que he contado no sirve de nada porque a quien tenga mediana idea del asunto de los juegos de caracteres del QL, de las impresoras, del código ASCII y demás, no le habrá aportado nada; y a quien no tuviese ni idea de todo eso creo que no le habrá ayudado mucho, tantos líos. El caso es que tal y como he contado he configurado un Quill en un disco aparte para escribir textos en inglés: la idea se me ocurrió al tener que mandar una colaboración al FIG, en la que debían aparecer los dichosos corchetes...

Marcos Cruz (QLave-242)
Madrid, Diciembre 1988

LA BATERIA DE VALENTE

A comienzos de año adquirí una batería de Valente para mantener el reloj del QL. Me reventaba la "estupidez" del trasto este, que cada vez se despertaba en algún día de Enero de 1961... No es algo importante, pero con el Toolkit II, que hace uso de la fecha a la hora de grabar los ficheros, era interesante saber cuando se habían grabado, con WSTAT, sin necesidad de poner el reloj en su punto cada vez que encendía el ordenador.

Recuerdo que había visto en el QL World de Ocubre de 1987 varios modelos de pilas y baterías para el reloj. Llamé a Valente a interesarme por la marca de la batería que vendían... ¡Y, después de hacerme esperar un momento, me dicen que no lo saben, "que no pone nada"! (¿?)

Me arriesgo, y les pido una. Cuando la recibí, me dí cuenta de inmediato de que se trataba de un montaje más o menos artesanal, con un par de hojas fotocopiadas donde se explicaba cómo instalarla. ¿Comprendéis la desfachatez al ir a mirar y salir con que "no pone nada"? (Aunque no es descartable la posibilidad de que esa persona en concreto no supiese nada del origen de la batería.) Cuando hablé con ellos después por otro motivo, les pregunté directamente si la batería las hacían aquí, y me respondieron que se las montaban en algún sitio de electrónica o algo así, creo recordar.

Si la batería funciona correctamente da igual de donde venga, pero me causó bastante mala impresión que me dijesen la primera vez que no sabían de qué marca era.

El caso es que la instalé. Había que soldar unos cuantos cables aquí y allá en ciertos puntos de la placa que aparecían descritos en un diagrama de las instrucciones. Incluso había que levantar una pata de uno de los chips y soldar a ella un cable.

Bueno, parece que aquello funcionó bien en principio. Después, la primavera fue desvaneciéndose en el éter y los calores veraniegos hicieron mella en el trasto: el encender el aparato era una lotería -primitiva además, nunca mejor dicho-. La proporción de las ocasiones en que la fecha estaba al día fue reduciendose según subía el termómetro. En un par de ocasiones de desesperación, destornillador en mano, abrí el sufrido QL ante la sospecha de que algún cable de la batería se hubiese desoldado, pero nada. Tomé nota de cuánto tiempo pasaba entre cada apagado y cada encendido, así como el tiempo de funcionamiento y el orden de apagado y encendido de los periféricos. Resultado: no parecía haber ninguna relación entre esos factores y la permanencia o no de la fecha.

Desesperado, me olvidé del asunto. Cada vez que encendía el QL, ponía el reloj en hora con SDATE y ya está (si por casualidad no hacía falta, daba un bote hasta el techo, claro).

Y he aquí que llegó el hermoso otoño y la batería volvió a funcionar. Increible. Es por esto que puedo decir que el problema está en la temperatura del QL, pues no hay otra explicación y las pruebas tienden hacia dicha conclusión. Ahora es rarísima -un par de veces en los últimos meses- la ocasión en que la fecha se corrompe al encender el aparato.

Me temo que cuando las nieves se derritan, el ciclo vital de la misteriosa batería de Valente volverá a comenzar...

CONVERTIDOR SERIE-PARALELO, TAMBIEN DE VALENTE

Un convertidor serie-paralelo tiene una entrada serie RS232 y una salida paralelo Centronics. Permite emplear una impresora con entrada Centronics con un ordenador con salida RS232, como es el caso del QL.

Este tipo de aparatos suelen tener cierta tendencia a fallar, a "comerse" o alterar caracteres. Eso me ocurrío con el que adquirí de Valente: después de unas semanas de uso, comenzó a "comerse" de vez en cuando retornos de carro y a cambiar espacios por caracteres gráficos. Me lo cambiaron por otro sin ningún

problema.

Al cabo de otras tantas semanas, al nuevo apartao le empezó a pasar lo mismo más o menos. Harto ya del asunto, lo abrí con un destornillador y observé varias resistencias variables. Entonces escribí un sencillo programilla que imprimiese una y otra vez el juego de caracteres completo, hasta el símbolo de copyright, cada vez que se pulsase una tecla. Así, girando en uno y otro sentido cada uno de las resistencias y viendo los resultados, se solucionó el problema. En realidad tuve bastante suerte, pues la primera de las tres resistencias que giré solucionó el asunto. Creo que, dado los problemas que estos adaptadores pueden dar -y dan, de vez en cuando-, esta nota servirá de ayuda a quien esté en el mmismo caso.

(Conviene anotar qué resistencia se ha movido y en qué sentido, para las siguientes veces.)

¿DISCOS DE SIMPLE CARA?

En teoría, un disco de simple cara tiene sólo magnetizada una de sus caras. La otra debería ser símplemente una superficie plástica sin ningún recubrimiento. Pero en la práctica no es así.

Este fenómeno de los discos de simple cara lo he observado en los discos de 3,5 pulgadas -en otros no lo he probado- en MSX, Spectrum y QL. Esto quiere decir que el "problema" está en realidad en los discos.

¿Cual es el "problema"? Sencillamente, que de simple cara nada de nada.

¿Comprobación? Atentos:

1. Cójase un disco de simple cara.
2. Formatéese como si nada, como si de doble cara se tratase.
3. Llénese de programas y pantallas hasta que no quepa nada más.
4. Cárguense los programas y pantallas uno a uno.
5. Compruebese que todos pueden leerse y que están íntegros.

¿Cómo es posible?

La primera pista de que algo no es correcto es el hecho de que el formateado pueda llevarse a cabo. Si una de las caras del disco no fuese magnetizable, el formateado daría error.

La segunda pista -definitiva- es llenar el disco (para que la segunda cara quede tammbién ocupada totalmente) y luego comprobar la integridad de lo grabado.

En alguna parte oí que lo de simple y doble cara es un timo, que lo que se hace es coger los discos de mayor calidad venderlos con la etiqueta de "doble cara"; y los de menor calidad como de "simple cara". No sé si eso es cierto, pero gato encerrado sí hay.

La Trump Card permite formatear discos a simple cara añadiendo un asterisco como undécimo carácter del nombre, dando entonces 720 sectores. Pero he cogido un disco -Philips y bastante viejecillo- que lleva explícitamente escrito "Single Sided" y lo he formateado normalmente a 1440 sectores. Lo he llenado de ficheros, hasta que dió error "Drive lleno", con un programa así:

```
A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
FOR N=0 TO 500
  N$=N
  OPEN_NEW#3,"FLP1_"&N$
  PRINT#3,A$
  CLOSE#3
END FOR N
```

Y he vuelto a leer todo con otro programilla así:

```
A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
FOR N=0 TO 500
  N$=N
  OPEN_IN#3,"FLP2_"&N$
  INPUT#3,B$
  AT 0,0:PRINT B$
  IF A$<>B$:PRINT "ERROR!":STOP
  CLOSE#3
END FOR N
```

¿Resultado? ¡Ningún error! Como si el disco fuese de doble cara.

Para mayor seguridad, los discos que tengo antiguos de simple cara los formateo a simple cara, pero no sería mala idea diseñar algún tipo de programa que leyese y escribiese mucho en la segunda cara de un disco para medir su grado de fiabilidad con el tiempo, una especie de banco de pruebas. Así, se probaría si los discos de simple cara son sólo "los deshechos".

El asunto, visto desde otro punto de vista, es muy serio. Podría presentarse una denuncia ante una asociación de consumidores. ¿Por qué? Vamos a ver: se supone que el disco sólo puede usarse por una cara, pero se puede usar por los

dos, luego ¿por qué se vende más barato? ¿en qué se diferenencia de los de doble cara? ¿dónde está el truco? ¿no será, como es fácil sospechar, que la calidad del recubrimiento magnético EN TODO EL DISCO es menor?

Borrad esa expresión de asombro de vuestros rostros y haced la prueba, ¡venga!

Marcos Cruz, (QLave-242)
Madrid, Diciembre 1988

INTENTANDO ADIESTRAR EL ESCARABAJO PELOTERO

Las rutinas de José Carlos de Prada para leer el estado del ratón de Investrónica (¡perdón, quise decir escarabajo pelotero!) permiten a uno diseñar sus propios programas para que funcionen con dicho engendro. No obstante, desde que topé en mi devenir con el susodicho ente repugnante, no he pretendido otra cosa que liberarle de su condición de inútil supremo. Hasta ahora no he tenido éxito en tamaña empresa.

Bueno, la idea era escribir una rutina en SuperBASIC para compilarla y dejarla como tarea residente. La función de la tarea sería leer el estado del escarabajo pelotero y engañar al QL para que creyese que en lugar del mísmo se han pulsado las teclas del cursor correspondientes. De este modo, con sólo EXECutar dicha tarea, el escarabajo podría usarse en cualquier programa, ¡incluso en Quill! Hasta aquí todo muy bonito, pero en la práctica no es así.

En primer lugar, ¿por qué, si el escarabajo pelotero se conecta a un port de joystick, no puede leerse con INKEY$? La respuesta la tendremos si ejecutamos el siguiente programa:

```
100 REMark Marcos 26/10/1988
110 REMark
120 CLS:TK2_EXT:REMark Toolkit II
130 PRINT "bits   01234567"
140 REPeat bucle
150   AT 1,0
160   FOR fila=0 TO 7
170     PRINT "fila"!fila!BIN$(KEYROW(fila),8)&CHR$(10);
180   END FOR fila
190 END REPeat bucle
```

Ejecutad el programa anterior con el escarabajo pelotero conectado y movedlo. Como observaréis, no se activa sólo el bit correspondiente a la tecla del cursor -o de tecla función, según donde esté conectado el bicho- correspondiente al movimiento, sino que llegan a activarse todos o casi todos los bits de igual peso de todas las filas... ¿? ¡Por eso no se puede leer el escarabajo con INKEY$!

¿Que se puede hacer, entonces? Se me ocurrió lo siguiente:

```
100 :
110 REMark Programa para simular el funcionamiento
120 REMark de un joystick
130 REMark con el ratón de Investrónica
140 REMark conectado a CTL1
150 REMark
160 REMark Marcos 28/10/1988
170 REMark
180 REMark Debe ser compilado y ejecutado como tarea
190 :
200 SET_PRIORITY 1:REMark para Turbo
210 IMPLICIT% peso_bit,bit,fila,n_filas,letra:REMark para Turbo
220 :
230 REPeat bucle
240   letra=0
250   IF peso_en_filas%(16)>1:letra=200
260   IF peso_en_filas%(2)>1:letra=192
270   IF peso_en_filas%(4)>1:letra=208
280   IF peso_en_filas%(128)>1:letra=216
290   IF peso_en_filas%(64)>1:letra=32
300   IF letra
```

```
310     PRINT "Letra= ";
320     REPeat espera
330       FOR fila=0 TO 7
340         IF KEYROW(fila):NEXT espera
350       END FOR fila
360     END REPeat espera
370     POKE_W 163978,0
380     PRINT letra
390   END IF
400 END REPeat bucle
410 :
420 DEFine FuNction peso_en_filas%(peso_bit)
430   n_filas=0
440   FOR fila=0 TO 7:IF KEYROW(fila)&&peso_bit:n_filas=n_filas+1
450   RETurn n_filas
460 END DEFine peso_en_filas
470 :
```

Si el programa se ejecuta -para hacer pruebas no hace falta compilarlo- aparecerá el código ASCII del carácter correspondiente al movimiento del escarabajo.

El funcionamiento se basa en averiguar si, para cierto peso de bit, están activadas varias filas del teclado. Si es así, es que el ratón se ha movido. Pero esto es todo lo que hay, no he podido ir más allá. Todo intento de simular que la tecla correspondiente se ha pulsado (en lugar de símplemente imprimir su código ASCII) ha sido infructuoso.

En el conjunto de procedimientos de ejemplo que viene con Turbo, hay uno que, compilado, hace que los caracteres "-" y "_" intercambien sus posiciones en la tecla en que están. Para ello detecta la pulsación de cada uno mediante INKEY$, en un bucle sin fin, y, si han sido pulsados, los sustituye por el otro mediante TYPE_IN. Pero si en vez de INKEY$ se usa KEYROW, el procedimiento no funciona.

En el libro de Adrian Dickens no he encontrado ninguna referencia a todo esto. El problema, lo que me falta saber cómo hacer, es en resumen lo siguiente: ¿Cómo, tras detectar cierta combinación de teclas mediante KEYROW, simular que se ha pulsado otra tecla cualquiera?

Marcos Cruz (QLave-242)
Madrid, Diciembre 1988

AGRADECIMIENTOS

Yo Nacho Enrique Cabero, Licenciado en Medicina de 26 años de edad, quiero agradecer a todos los que he conocido alrededor del QL y a mi ordenador Sinclair QL el que ahora este trabajando como Técnico de Sistemas en el Area de Sanidad en C.C.S. en Madrid.
El QL es aún mi primer ordenador y espero que por muchos años y con el aprendí a programar primero en Basic luego ensamblador.. pascal.. archive.. c.. bcpl.. y así un porrón de lenguajes

Ahora trabajo con miniordenadores tipo VAX con un lenguaje de base de datos para hospitales llamado MUMPS, pero mis fundamentos de programación los aprendí en el QL y de gente que trabajaba con el QL. Insisto:

Gracias a todos vosotros

Nacho Enrique Cabero
Nueva dirección en Madrid
Miguel de la Iglesia 3 1º
28027-Madrid
Telfno (91) 2683234

Una vez lanzado el manifiesto, y referida mi dirección advertir que estoy abierto a cualquier sugerencia del personal de Madrid así como de el de Valladolid al que vuelvo bastantes fines de semana.

Desde mis últimas intervenciones en QLAVE han cambiado muchas cosas incluida mi

orientación profesional, que se ha matizado a caballo de la Informática y la Medicina. Ahora trabajo entre ordenadores 8 horas al día y aunque me cuesta bastante, procuro no descolgarme del QL al que insisto, le debo mucho. Si no me he descolgado más, ha sido gracias a Salvador y sus desvelos por involucrarme en sus locuras particulares que está absolutamente locuelo por el QL, casi más que yo.

Pese a todo en estos meses he encontrado tiempo para seguir aprendiendo cosas con el QL las cuales os paso a relatar a continuación.

En mi biblioteca particular desde el verano, han aparecido los siguientes títulos:

-Programación en COBOL estructurado L.R.Newcomer McGraw-Hill
-Programación avanzada en COBOL estructurado L.R.Newcomer McGraw-Hill
-COBOL y sus aplicaciones en los negocios S.Philippakis McGraw-Hill
-Programación en LISP C.Queinnec
-APL enfoque interactivo L Gilman ILMUSA
-Teleinformática C.Macchi OMEGA
-Lenguajes de Programación Diseño e implementación T.W.Pratt McGraw-H
-Advanced Forth A.J.Reynolds SIGMA press
-Assembly and assemblers /the MC68000 family G.W.Osborne Prentice Hall

Son ocho libros,ocho los cuales no he leído aún en su totalidad pero sí lo suficiente como para tener una impresión que poderos brindar al respecto.

De estos ocho libros cuatro están dedicados a un proyecto mío particular que habrían de juntarse con otros cuatro libros que ya tenía para dar un resultado aún en el limbo, y que sólo podre acabar si tengo el estímulo de gente como vosotros ya que desde éstas líneas voy a intentar compartirlo con todos vosotros. Se trata de desarrollar un compilador de COBOL para QDOS con linkado en tiempo de ejecución aprovechando los recursos multitarea del QDOS. El proyecto actualmente está desarrollado en una cuarta parte más o menos pero dado que lo empecé en Abril y sólo pude trabajar de continuo hasta mediados de Junio ahora estoy un poco descolgado de el. Para volverlo a retomar tengo que volver a hacer un análisis para ponerme en situación, aparte que tengo que hacer algunas modificaciones de base que ahora es cuando se ven de modo que el reflejar el proyecto en éstas líneas no va a hacer sino ayudarme a llevarlo a buen término.
A medida que se vaya exponiendo el compilador en él se admitirán todas las sugerencias que desee el personal hacer siempre que sean justificables.

Pero de todo ésto hablaremos más tarde ahora vamos a comentar algunos de los libros antes mencionados prescindiendo del COBOL y su implementación.

De todos los libros hay uno que destaca de sobremanera:
ASSembly and assemblers: The Motorola MC68000 Family de G.W.Gorsline es una auténtica maravilla, con dos únicas pegas: vale más de 8000 pesetas, y está en inglés. Si ya en un artículo de QLAVE hacía una revisión de libros de ensamblador 68000 y de ella me quedaba con el del Grupo Waite, después de ver éste último, señores me quedo con él, aunque muchas de las cosas novedosas que en él se encuentran no son aplicables al QL.
Destacable son:
-estudio minucioso y claro de una buena programacion ESTRUCTURADA en ensamblador
-Estudio claro de el aprovechamiento de recursos de sistema operativo en general, desde ensamblador
-Estudio claro de los procesadores avanzados de la familia (68010,68012,68020) no tan retorcido como en el libro del Grupo Waite
-Estudio claro de el control de los coprocesadores de la familia tanto del de coma flotante como de los de acceso directo a memoria(DMA) y manejo general de memoria (MMU), desde la CPU 68XXX así como microprogramación extendida de la CPU (tabla de vectores) estructura de buses y memoria Caché
En resumen, el libro creo que no tiene un pero que poner a ninguna de sus seiscientas y pico páginas recomendandolo a todo el personal

El segundo libro que voy a comentar en ésta ocasión (el resto los comentaremos en otras ocasiones) es el APL enfoque interactivo de Leonard Gilman y Allen J.Rose editado por LIMUSA -Mexico.
A este libro llegue, como a muchas otras cosas, por culpa de Salvador Merino, y se lo agradezco.
El APL es un lenguaje muy curioso y potente. Pese a estar interpretado su ejecución seudocompilada es rápida y efectiva, lo cual junto a su especial

manera de ver los datos hace que muchas cosas se hagan extremadamente sencillas cuando en otros lenguajes serían poco menos que labores de locos. Además, los poseedores del QL disponemos de una versión del lenguaje casi casi casi perfecta y en algunas otras cosas muy extendida.
La sintaxis APL es un poco complicada pero mas sencilla que la del FORTH teniendo una especial facilidad en el manejo de conjuntos de cifras así como en la ejecución de programas que se pueden modificar sobre la marcha mediante el comando de ejecución (cosa que además del APL sólo permite hacerlo el MUMPS que yo sepa).
El libro de Gilman y Rose que ha llegado a mis manos es la tradución al mejicano de la segunda edición en inglés, aunque existe una tercera édición no he podido localizarla.
En su exposición es muy metódico y correcto pareciendo más una relación de clases magistrales que un libro de texto, siendo poco amable en las disertaciones, ahora, eso sí no se les escapa nada PARA EL APL DEL IBM 5100 lástima que éste es bastante anticuadillo,no obstante es perfectamente tranportable en su mayoría al APL del QL. El libro hace incursiones por todos los aspectos del APL desde su gestón en un entorno mutiusuario hasta su gestión de base de datos particular, entreteniendose un poco más en la descripción de cálclos vectoriales, la especialidad del APL.
Para poderse enfrentar de un modo racional a éste seudo-ladrillo sería conveniente empezar por algo más llevadero como podría ser el divertidísimo libro: "APL lenguaje para programadores diferentes" de la enciclopedia de la informática aplicada de ediciones siglo cultural, escrito por Juaan Ruiz de Torres, una de esas coleciones de libros de bolsillo que tanto se estilan.

Para no cansaros, el comentario de los demás libros lo dejo para otros meses así como un comentario más profundo sobre el APL.
Nacho

IECSOFT QL COBOL VS 1.00

Aquí comienza una larga serie de artículos cuyo objetivo es el desarrollo de un compilador COBOL para el QL. Como no hay más remedio empezaremos por lo que exigen las normas internacionales.

IECSoft QL.COBOL por NACHO ENRIQUE (C) 1988 V$ 1.0

El presente compilador de COBOL es una versión del COBOL ANS X3.23-1985 desarrollado específicamente para trabajar en el entorno de los sitemas operativos QDOS(R) y ARGOS(R) para el microprocesador Motorola 68000.

Tanto el compilador como este manual son propiedad del autor y bajo ningún concepto se pueden realizar cualquier tipo de reproducción total o parcial de dicho material fuera de los cauces legalmente autorizados por éste.

En ninguna circustancia IECSoft se hace responsable de cualquier daño, pérdida de información o perjuicio económico ocasionado por la utilización de éste compilador.

Para información y guía del usuario a continuación se extracta el "Goverment Printing Office Form Number 1965 - 0795689":
«Cualquier organización interesada en reproducir total o parcialmente el informe y especificaciones COBOL, o en utilizar las ideas del informe como base para manuales de instrucciones o cualquier otro propósito, puede hacerlo libremente. Sin embargo, todas estas organizaciones están obligadas a reproducir ésta sección en la introducción del documento. Quienes utilicen un pasaje corto, están obligados a mencionar "COBOL" para identificar la fuente, pudiendo no trancribir toda ésta sección.
COBOL es un lenguaje industrial y no es propiedad de ninguna compañía o grupo de compañías ni tampoco de ninguna organización o grupo de organizaciones.
No se otorga ninguna garantía expresa ni implícita por ningún contribuyente ni por el Comité COBOL sobre la precisión y funcionamiento del lenguaje de programación. Más aún, no se asume responsabilidad por ningún contribuyente ni por el Comité relacionada con lo anterior.
Se han establecido procedimientos para el mantenimiento de COBOL. Las preguntas y proposiciones de modificaciones deben dirigirse al "Executive Committee" de la "Conference of Data Systems Languages".
Los autores y propietarios de los derechos del material usado aquí son:
FLOW-MATIC (Trademark of Sperry Rand Corporation). Programming for the UNIVAC(R)

I and II, Data Automation Systems copyrighted 1958,1959 by Sperry Rand Corporation; IBM Commercial Translator, Form No. F28-8013, copyrighted 1959 by IBM; FACT, DSI 27A5260-2760 copyrighted by Minneapolis- Honeywell

ellos han autorizado específicamente el uso de éste material, total o parcialmente en las especificaciones de COBOL. Tales autorizaciones se extienden a la reproducción y uso de las especificaciones de COBOL en manuales de programación y publicaciones similares.»

Prologo:

Cuando en Febrero de 1988 tomé el bolígrafo y dibujé el organigrama del programa principal de éste compilador, se planteaba ante mí el gran reto de llevar a buen puerto éste proyecto intentando tener siempre en mente las exigencias de posibles usuarios del mismo. La tarea es árdua pero creo que el resultado merece la pena.

El lenguaje COBOL es tan antiguo como el más antiguo de los lenguajes de alto nivel, pero sin embargo su utilización generalizada y su cuidada puesta al día hacen que no haya perdido ni un ápice de frescura en su dilatada historia. Bien es verdad que las estructuras de control de flujo de programa de la revisión ANS X.23-1985 poco tienen que ver con los esquemas de flujo del COBOL de los 60, pero el COBOL de los 80 sigue manejando complicadas estructuras de datos con la misma naturalidad de siempre, y es en esta naturalidad en el manejo de estructuras de datos donde sigue residiendo el atractivo de éste lenguaje en el mundo de la informática de gestión.

El presente manual no pretende enseñar a programar en COBOL, para ello existen libros, como los que aparecen al final de éste manual en la Bibliografía, escritos por personas especializadas en la materia. Aunque éso sí én el se encontrarán descritas, todas las facilidades de éste lenguaje de programación.

Se ha hecho un gran esfuerzo en aprovechar al máximo las facilidades que brinda el QDOS, a fín de no hechar de menos ninguna cualidad del mismo, incluso algunas funciones COBOL se han adaptado para aprovecharlas, así, por ejemplo, el uso de la LINKAGE SECTION y la función CALL ,trabajan mediante la activación y suspensión de programas en multitarea. Asimismo se ha intentado en lo posible generar programas objeto lo más compactos posible siendo en general de código relocalizable en todo momento, aún cuando el código fuera mayor de 64Kbytes para ello ha habido que utilizar, para el compilador una mezcla de código escrito en

C y en ensamblador, mezcla fácilmente realizable gracias al compilador QLC de GST Computer Systems Limited.

Espero que este compilador sea del agrado de todos, en todo caso, cualquier sugerencia respecto a posibles mejoras en futuras versiones siempre será bien recibida.

Nacho Enrique

Este es el planteamiento inicial y lo llevaremos a buen térmmino si vosotros quereis, si no... comunicádmelo un saludo.

LET'S GO FORTH
(Capítulo tercero, a pesar de todo)

FIG y Forth Dimensions.

He recibido los cuatro primeros ejemplares del volumen X de Forth Dimensions. En verdad se han dado prisa, en el FIG, en responder a mi solicitud, cosa muy de agradecer (Thanks, folks!). No sé exactamente si el envío se ha producido desde USA o desde Inglaterra, pues aparte de la dirección del FIG en USA impresa en el sobre, no hay indicación ni sello alguno; al menos el sobre llevaba una nota indicando que en caso de devolución se enviase a la dirección del FIG en Inglaterra.

El Forth Dimensions aparece cada dos meses, comenzando cada nuevo volumen con el número de Mayo/Junio. Así, he recibido en este primer envío hasta el número de Noviembre/Diciembre, y una tarjeta de socio -para firmar y todo- que como curiosidad es interesante, pero que me parece que por aquí no tendrá mucha utilidad.

La impresión inicial es muy favorable: una publicación de aspecto profesional, muy cuidada, de poco menos de 40 páginas tamaño cuartilla, a tres columnas. La variadad de temas tratados la hace muy interesante: programas, rutinas, comentarios, cartas.

Hay dos "pegas", no obstante:

1. Todos las referencias al hardware son para PC o Mac, bastante lógico y bastante inevitable (¿Sabrán allí lo que es un QL?)

2. Es dificil evitar una envidia cochina al ver la cantidad de seminarios, conferencias y demás "meetings" que se montan allende los mares en torno al Forth... es como si estuvieses en una isla desierta y te mandasen cada dos meses el folleto de una pastelería.

De vueltas con el SISIFOrth:

Espero que a estas alturas habréis recibido el fichero SISIFOrth_scr con el código fuente de SISIFOrth para el Forth de Computer One. Espero que los Forth maniacos lo hayáis probado. Si a alguien se le ocurre algo para mejorarlo, que nos lo cuente sin dudarlo (un pareado me ha quedado). Sobre todo, pensad en algún sistema experto interesante (por ejemplo, para reparar reactores nucleares o QLs, o para encontrar petróleo).

Lo único que quería añadir era esto: ¿Por qué el nombre de SISIFOrth? Quien tenga alguna idea de mitología griega habrá sospechado ya de qué iba el título, pero, por si acaso, ahí va la explicación: Sísifo fue condenado por Zeus a subir una enorme piedra a la cima de un monte, pero al llegar arriba la piedra siempre se le caía y el pobre diablo tenía que volver a bajar, y así eternamente una y otra vez... ¿Lo cogéis? Pues eso.

Un par de palabrillas para pensar:

El fig-Forth carece de las palabras DEPTH y PICK (no creo que haga falta

indicar qué hacen, ¿no? ¿duyus pikinglis?). No recuerdo si sus definiciones las saqué yo mismo o las vi por ahí en alguna parte, pero las he encontrado en un pedazo maloliente de pergamino que ha salido de un rincón polvoriento...

```
: DEPTH ( --número_elementos_pila)
  SP@   ( dirección_tos)
  S0 @  ( dirección_tos,dirección_bos)
  -     ( -bytes_pila)
  -2    ( -bytes_pila,2)
  /     ( número_elementos_pila)
;
: PICK ( número_elemento--elemento)
  2 *  ( número_byte-2)
  2 +  ( número_byte)
  SP@  ( número_byte,dirección_tos)
  +    ( dirección_elemento)
  @    ( elemento)
;
```

Jupiter Ace.

¡El único ordenador conocido -por mí- que incorpora el Forth en ROM como estandar en lugar del BASIC!

Exteriormente es muy similar al ZX-81, pero en color blanco. Mejor cabría decir que se parece al primitivo ZX-80, que era blanco. Las teclas son similares a las del Spectrum, de goma; de hecho, los diseñadores del Jupiter Ace habían formado parte del equipo de diseño del ZX Spectrum.

El parecido con el ZX-81 también se nota en sus características:

Dimensiones:	215 x 190 x 30 mm
Peso:	245 g
Microprocesador:	Z80-A
Velocidad de reloj:	1 MHz
Memoria ROM:	8 Kbytes distribuidos en 2 EPROM de 4 Kbytes
Memoria RAM:	3 Kbytes ampliables externamente a 51 Kbytes
Visualización:	Blanco y negro 32 x 22 caracteres 64 x 48 de resolución en modo gráfico
Interfaces:	Salida para televisión Entrada/Salida para cassette Entrada alimentación eléctrica 9 v Dos conexiones a la placa para ampliación
Teclado:	QWERTY de 40 teclas móviles moldeadas sobre una lámina de plástico, muy similar al del Spectrum
Incluye:	Transformador para alimentación eléctrica Cable para conectar al televisor Cable para conectar al cassette Manual de 180 páginas

Entre las características más peculiares, cabría citar las siguentes:

* La interface de cassette es muy fiable.
* La documentación suministrada es de muy buena calidad y constituye una buena introducción al Forth
* El Forth incorporado no es Forth-79 ni fig-Forth, sino que tiene algunas peculiaridades.
* Por medio de un adaptador, puede emplear los famosos RAM-packs del ZX-81, tanto de 16 como de 48 Kbytes, así como la mayoría de las interfaces Sinclair para dicho ordenador.

(Datos obtenidos de la enciclopedia "mi computer", página 150)

Resumen:

El Jupiter Ace constituyó en su momento -1983 más o menos- una valiente iniciativa y fue un completo pionero. Como la mayoría de las veces ocurre, quien se arriesga a ir en contra de la coorriente -en este caso, quien pone en ROM el Forth en vez del Basic- no tarda en ser destinado a la incomprensión y al olvido. Puede decirse que en Inglaterra gozó de alguna inicial aceptación entre los aficionados al Forth, pero con la difusión de otros ordenadores más sofisticados (sobre todo a nivel de memoria y gráficos) -léase Spetrum- se hundió en los confines del baúl de los recuerdos. En España es prácticamente desconocido, y ni siquiera sé si estuvo aquí a la venta alguna vez; lo distribuía la empresa Sushiro Data.

He dicho "es" en lugar de "era", "tiene" en lugar de "tenía", etcétera, al referirme a las características del Jupiter Ace porque, a pesar de que esté definitamente muerto desde hace mucho tiempo, se merece esta pequeña consideración, snif, snif...

Marcos Cruz, (QLave-242)
Madrid, Diciembre 1988

ALARMA INFALIBLE

Este sencillo programilla convierte al QL en un preciso despertador dotado de una "agradable" melodía. Quien no se despierte con esto es que está muerto:

```
100 REPeat hora
110   fecha$=DATE$
120   hora$=fecha$(13 TO 17)
130   AT 0,0;
140   PRINT hora$
145   IF hora$="09:30":EXIT hora
150 END REPeat hora
160 :
170 REPeat alarma
180   IF INKEY$<>"":EXIT alarma
190   BEEP RND(1 TO 32767),RND(0 TO 255),RND(0 TO 255),RND(-32768 TO 32767),
      RND(-32768 TO 32767),RND(0 TO 15),RND(0 TO 15),RND(0 TO 15)
200   REPeat pausa:IF NOT BEEPING:EXIT pausa
210 END REPeat alarma
```

Lo primero es poner el reloj en hora con SDATE, y después alterar la línea 145 para poner en vez de "09:30" la hora y minutos en que debe sonar la alarma, en el formato "hh:mm". Cuando suene, bastará pulsar con suavidad una tecla para enmudecer al QL, aunque es posible que prefiráis desenchufarlo de un manotazo. (Probad un GOTO 170 para escuchar la versión QL de un canto gregoriano.)

El programa podría fácilmente perfeccionarse para pedir las horas con INPUT y además compilarse para ejecutarse como tarea residente, de modo que funcione mientras hacemos otras cosas en el aparato. Esa labor queda para el que se anime.

El Toolkit II tiene una función de alarma, pero apenas suena un pitido escuálido. Esto es otra cosa.

¡No olvidéis dejar el QL encendido toda la noche, que si no no funciona!

Marcos Cruz (QLave-242)
Madrid, Diciembre 1988

LIBRO : MATEMATICAS PARA PROGRAMADORES
AUTOR : WILLIAM BARDEN, Jr.
Titulo original : MICROCOMPUTER MATH
EDITORIAL : Ediciones Anaya Multimedia, S.A.
Precio : 848 ptas.
Páginas : 160

Este libro lo leí en la playa en julio 88 tostándome al sol (aún no era socio de QLave). El libro es sencillo y muy divertido (parece dedicado a un BEBE).

El objetivo de matemáticas para programadores es poner fin a algunos de los misterios que rodean las operaciones matemáticas especiales que se emplean en Basic y en lenguaje ensamblador. Y aunque este libro iba dirigido a los microprocesadores Z80,8080,6502 y 6800, sirve de base para los principiantes del MC 680XX, pues los misterios siguen siendo los mismos.

¿ Qué va a encontrar ?

- El sistema binario : donde empieza todo.... Big Ed aprende binario. Más sobre bits, bytes y binario. Paso de binario a decimal. Paso de decimal a binario. Rellenar a ceros hasta ocho o dieciséis bits.
- Octal, hexadecimal y otras bases numéricas... El chile está bien en Casiopea. Hexadecimal. Octal. Trabajando con otras bases numéricas. Convenios estándar.

- Números con signo y notación en complemento a dos .... Big Ed y el bínaco. Sumar y restar números binarios. Representación en complemento a dos. Extensión del signo. Suma y resta en complemento a dos.
- Acarreos, errores de desbordamiento e indicadores....... Este restaurante tiene una capacidad de +127 personas. ¡Evitando errores de desbordamiento! Errores de desbordamiento. Acarreo. Otros indicadores. Indicadores en los microordenadores.
- Operaciones lógicas y desplazamientos....... El enigma británico. Operaciones lógicas. Operaciones de desplazamiento.
- Multiplicación y división....... Zelda aprende cómo desplazar por sí misma. Algoritmos de multiplicación. Algoritmos de división.
- Multiple precisión....... ¿Tienen algo que ver las series de Fibinacci con la televisión? Suma y resta empleando múltiple precisión. Multiplicación en múltiple precisión.
- Fracciones y factores de escala ... Big Ed pesa los números. Fracciones en sistema binario. Operando con fracciones en sistema binario.
- Transformaciones ASCII..... Big Ed y el inventor de los códigos ASCII. Paso de ASCII a enteros binarios. Paso de ASCII a fracciones binarias. Paso de enteros binarios a ASCII. Paso de fracciones binarias a ASCII.
- Números en punto flotante..... ...y tres mil platos combinados para la nave nodriza... Notación cientifica en punto flotante. Uso de potencias de dos en lugar de potencias de diez. Números en punto flotante de doble precisión. Cálculos en los que se emplean números binarios en puntos flotantes.
- Y muchisimos ejercicios y soluciones.

Como podreis observar, hay muy poco desperdicio en este pequeño libro.

No tenia pensado comentar este libro, pero debido a que los conocimientos en que basa el capitulo IX de Practicando con el Superforth los aprendí hace dos años y medio con este libro, he decidido comentarlo (o darlo a conocer).

Según mi opinión, es quizás el mejor libro sobre aritmética binaria que se ha editado en Castellano. No deberia faltar en la biblioteca de ningún programador.

Salvador Merino
Fuengirola (Málaga)
Diciembre 1988.

PRACTICANDO CON EL SUPERFORTH (Capitulo IX)

Cuando estaba tan emocionado escribiendo el capitulo VIII a finales de Abril, no creia ni por lo más remoto que iba a pasar lo que hemos visto todos. Serafín era aún el presidente y yo confiaba en él, luego Diego se hace presidente a la fuerza nombrandose el mismo sin preguntar a los socios (Ver estatutos del Club 1986, y vereis que el articulo 11 dice "los socios podrán ser sancionados por la junta directiva por infrigir reiteradamente los presentes estatutos o acuerdos de la asamblea general o de la junta directiva.......". Me temo que el señor Diego nunca ha leido los estatutos del Club.).

Cuando recibí el boletín Marzo/Abril 1988 en Junio, y ví que Serafín ya no era el presidente ya me temia lo peor, y más aún cuando ese mismo mes Rafael Candau me contó por teléfono lo que ocurrió el dia 13 de Abril de 1988 en Zaragoza (una historia lamentable digna de ser censurada). Pasaron los meses y todas las advertencias de Rafael se hicieron realidad (Diego no sabe programar. Promete mucho, pero no cumple. Tratar con él es desesperante. Creo que solamente le interesa nuestro dinero).

Cuando recibí el boletín Mayo/Junio 1988, pude observar que no habia ninguna colaboración de Zaragoza y que tenia menos páginas habiendo un gran stock de colaboraciones mias y de otros socios amigos mios.

Ahora a finales de Diciembre 88, me he visto obligado a crear un engendro llamado CUQ. No sé si podrá sustituir lo que una vez fue QLave, pero podrá servir de salvavidas mientras se nos ocurre otra idea (o será permanente hasta que nuestros QLs exploten).

Dejemonos de rollo y vamos al grano. En el último capitulo se me ocurrió anunciar un pequeño proyecto que tenia entre manos, pero debido a las malas noticias de la gestión de Diego, se fue quedando en el baul de los recuerdos (estaba francamente desmoralizado).

Antes de continuar, otra interrupción. Yo igual que otros tengo que agradecer a todos los que han hecho realidad una idea como la que fue QLave en el momento de su fundación (una copia del Club Quanta Inglés a la Española). Si no es por vosotros, yo nunca me molestaria en escribir un texto o un programa que nadie va a leer o usar alguna vez. Pero lo más importante de todos es que VOSOTROS OS AYUDAIS UNOS A OTROS A NO SENTIRSE SOLO O OLVIDADO.

El tema de este capitulo y los siguientes (es muy largo) es una extensión del Superforth para trabajar con números enteros de 64 bits. Un número de 64 bits tiene un rango con signo de -93223.3722032.7591841.344 a 93223.3722032.7591841343. Un número lo suficientemente alto como para poder usarlo en nuestra vida diaria en ptas, ya que los números de 32 bits en ptas se están quedando pequeños (¿ Hay alguien que no esté de acuerdo ?).

```
* Aritmética de 64 bits v1.0.
* Salvador Merino (Diciembre 1988). MERINO-SOFT ©

* This is an example of assembler source code which is assembled by the
* METACOMCO assembler or similar into a binary code file, which is loaded
* by the SUPERFORTH word LOAD_BIN into the SUPERFORTH dictionary.
* The format in the examples below must be followed to ensure that the
* correct number of bytes at the head of each word is present. This
* header is :
*
*          $4AFB          a marker ( word )
*          name_length    number of characters ( byte )
*          name           in ASCII
*          ( padding )    0 or 1 byte, only if length of name is even, to
*                         ensure the following code is on a word boundary.
*
* This header is created using the "code" macro which must be used in the
* form
*             code     'name',label
*              .          .              assembler instructions
*              .          .
*             next                       or a branch to a next
* label
*          the label is essential so that LOAD_BIN knows how much code to
*          load into a parameter field.
*
* Note that position independent code must be generated, you can refer to
* labels in other code definitions since the number of bytes in the
* above header is exactly the same as in the SUPERFORTH header itself.
* Each code word must end in a "next" macro or a branch to a "next" macro
* See chapter 11 of the SUPERFORTH manual.
*
* A 0 must be present at the end of the file to mark the end of the file,
* this is generated below by the "end_file" macro.
*
* Assemblers other than Metacomco's can be used in a similar way but the
* syntax for the macros will be different.
*
*
*
marker        equ      $4AFB             To mark the start of a code word
*
* macros
*
next          macro                      Fetches and executes the next
              move.w   (a1)+,d1          SUPERFORTH word, must be executed
              movea.w  0(a2,d1.w),a5     at the end of a code word
              jmp      0(a2,a5.w)
              endm
*
*
code          macro                      Creates a header
              dc.w     marker            Marks the start of a word
              dc.b     strg_end\@-strg_start\@     Gives the string length
```

```
strg_start\@   dc.b     \1                  The primitive's name
strg_end\@     equ      *
               cnop     0,2                 Pads to an even address
               dc.w     \2-*-2              Gives the code length
               endm
*
end_file       macro
               dc.w     0                   Must be placed at end of the code
               endm
*
*
******************************************
*          ARITMETICA DE 64 BIT          *
******************************************

* Negar un número de 64 bit

          code      'NEGAR',fin_negar
          move.w    d2,-(a3)
          neg.l     2(a3)
          negx.l    (a3)
          move.w    (a3)+,d2
          next
fin_negar

* Sumar dos números de 64 bit

          code      'SUMAR',fin_sumar
          move.w    d2,-(a3)
          move.l    (a3)+,d4
          move.l    (a3)+,d3
          move.l    (a3)+,d0
          move.l    (a3)+,d1
          add.l     d3,d1
          addx.l    d4,d0
* quizás sea lógico comprobar acarreo
          bcs       acarreo
          move.l    d1,-(a3)
          move.l    d0,-(a3)
          move.w    #1,d2
          bra.s     sumado
acarreo
          moveq     #0,d2
sumado
          next
fin_sumar

* Restar dos números de 64 bit

          code      'RESTAR',fin_resta
          move.w    d2,-(a3)
          move.l    (a3)+,d4
          move.l    (a3)+,d3
          move.l    (a3)+,d0
          move.l    (a3)+,d1
          sub.l     d1,d3
          subx.l    d0,d4
* Quizás sea lógico comprobar acarreo
          bcs       acarreo1
          move.l    d3,-(a3)
          move.l    d4,-(a3)
          moveq     #1,d2
          bra.s     restado
acarreo1
          moveq     #0,d2
restado
          next
fin_resta

* Multiplicar dos números de 32 bit dando un resultado de 64 bit
```

```
           code      'MULTIPL',fin_mult
           move.w    (a3)+,d0
           move.w    (a3)+,d3
           move.w    (a3)+,d4
           move.w    d3,d5
           move.w    d4,d6
           mulu      d0,d4
           mulu      d0,d3
           mulu      d2,d6
           mulu      d2,d5
           move.w    d4,-(a3)
           move.w    #0,d4
           swap      d4
           add.l     d4,d3
           move.w    d6,d0
           move.w    #0,d6
           swap      d6
           add.l     d6,d5
           move.l    d3,-(a3)
           move.l    #0,-(a3)
           move.w    d0,-(a3)
           move.l    d5,-(a3)
           move.w    #0,-(a3)
           move.l    (a3)+,d4
           move.l    (a3)+,d3
           move.l    (a3)+,d0
           move.l    (a3)+,d1
           add.l     d3,d1
           addx.w    d4,d0
           move.l    d1,-(a3)
           move.w    d0,d2
           next
fin_mult

* dividir un número de 64 bit por uno de 32 bit dando un resto de 32 bit y
* un cociente de 32 bit

           code      'DIVISION',fin_division
           move.w    d2,-(a3)
           tst.l     (a3)
           beq       cero0
           move.l    (a3)+,d0
           move.w    #0,d2
           move.w    #31,d4
           move.l    #0,d1
           move.l    (a3)+,d5
           move.l    (a3)+,d6
bucle0
           move.l    #0,d3
           asl.l     #1,d6
           roxl.l    #1,d5
           roxl.b    #1,d3
           tst.b     d3
           beq       limpio
           sub.l     d0,d5
           subx.b    d2,d3
           tst.b     d3
           beq       medio_bien
limpio
           cmp.l     d0,d5
           blt       menor
           sub.l     d0,d5
medio_bien
           cmp.l     d0,d5
           bhi       mal
           asl.l     #1,d1
           addq.l    #1,d1
           bra.s     dividido
menor
```

```
          asl.l     #1,d1
dividido
          dbf       d4,bucle0
          move.l    d5,-(a3)
          move.l    d1,-(a3)
          move.w    #1,d2
          bra.s     hecho
cero0
          move.w    #16,-(a3)
          move.w    #0,d2
          bra.s     hecho
mal
          move.w    #-18,-(a3)
          move.w    #0,d2
hecho
          next
fin_division

* dividir un número de 64 bit por uno de 16 bit dando un resto de 16 bit y
* un cociente de 64 bit

          code      'DIV',fin_div
          move.w    d2,d1
          tst.w     d2
          beq       cero
          move.w    #2,d2
          move.w    (a3)+,d3
          swap      d3
bucle
          move.w    (a3)+,d3
          divu      d1,d3
          move.w    d3,-(a4)
          dbf       d2,bucle
          swap      d3
          move.w    d3,-(a3)
          move.w    (a4)+,-(a3)
          move.w    (a4)+,-(a3)
          move.w    (a4)+,-(a3)
          move.w    #0,-(a3)
          move.w    #1,d2
          bra.s     hecho
cero
          adda.l    #8,a3
          move.w    #0,d2
hecho2
          next
fin_div


               end_file

               end

------------------------------------------------------------------------

: 2LNEGATE DEPTH 4 >= IF NEGAR
                         ELSE 6 ERROR THEN ;

: 2L+ DEPTH 8 >= IF SUMAR IF ELSE 2DROP 2DROP -18 ERROR THEN
                 ELSE 6 ERROR THEN ;

: 2L- DEPTH 8 >= IF RESTAR IF ELSE 2DROP 2DROP -18 ERROR THEN
                 ELSE 6 ERROR THEN ;

: 2L* DEPTH 4 >= IF MULTIPL
                 ELSE 6 ERROR THEN ;

: 2L/MOD DEPTH 6 >= IF DIVISION IF ELSE ERROR THEN
                    ELSE 6 ERROR THEN ;
```

```
: 2LM/MOD DEPTH 5 >= IF DIV IF ELSE 16 ERROR THEN
                    ELSE 6 ERROR THEN ;


VARIABLE BIT_SIGNO

: L#
BASE @ M/MOD >R >R
BASE @ UM/MOD >R
BASE @ UM/MOD SWAP 9 OVER < IF 7 + THEN
            48 + HOLD R> R> R> ;

: L#S   L#
2DUP >R >R OR >R
2DUP OR R> R> R>
2SWAP OR 0= IF ELSE RECURSE THEN ;

: L#> 2DROP #> ;

: 2L. DEPTH 4 < 6 ?ERROR DUP >R DUP 0< IF 2LNEGATE THEN <# L#S R> SIGN L#>
                        TYPE SPACE ;

: DENTRO? >R OVER <= SWAP R> <= AND ;

: DIGITO 48 - BASE @ 1-  OVER < IF 7 - THEN DUP 0 BASE @ 1- DENTRO? ;

: COMPRUEBA_SIGNO BIT_SIGNO @ IF 2LNEGATE THEN ;


: CONVIERTE
     1+ DUP >R C@ DUP 45 = IF 1 BIT_SIGNO ! DROP
           ELSE DIGITO IF  0 0. 2L+
               ELSE DROP COMPRUEBA_SIGNO 0 BIT_SIGNO ! R> EXIT THEN THEN


            BEGIN
         R>  1+ DUP >R C@ DIGITO IF  >R >R >R >R BASE @ UM* 0
           R> BASE @ UM* D+ 0
           R> BASE @ UM* D+ 0
           R> BASE @ UM* D+ DROP
           R> 0 0. 2L+
              ELSE DROP COMPRUEBA_SIGNO 0 BIT_SIGNO ! R> EXIT THEN
                                             0 UNTIL ;

: ENTRAR ." Introducir un número  "  PAD 80 EXPECT 0. 0. PAD 1-
                                                        CONVIERTE DROP ;

: L$ 0. 0. 32 WORD CONVIERTE  DROP ;

END_FILE
```

Definiciones :

2LNEGATE .- Niega un número de 64 bits.
2L+ .- Suma dos números de 64 bits.
2L- .- Resta dos números de 64 bits.
2L* .- Multiplica dos números de 32 bits dando un resultado de 64 bits.
2L/MOD .- Divide un número de 64 bit por uno de 32 bits dando un resto de 32 bits y un cociente de 32 bits.
2LM/MOD .- Divide un número de 64 bits por uno de 16 bits dando un resto de 16 bits y un cociente de 64 bits.
2L. .- Muestra el número de 64 bit que está en TOS.
L$.- Convierte una cadena de caracteres en un número de 64 bits dejandolo como TOS.

Ya sé que hay más definidas, pero estas ocho son las más importantes (las

demás sirven para definir las más importantes).

No he tenido suficiente tiempo para probar el buen funcionamiento de todas las nuevas definiciones debido a que solamente tenia en mente el terminar el boletín CUQ 4 antes del dia 31 de diciembre de 1988 (¡ Lo he conseguido por los pelos antes de las doce campanadas !). Pero no hay que preocuparse ya que es la primera versión sin depurar y optimizar, y la primera parte de un montón de definiciones que formarian entre todas ellas el paquete de 64 bits (cuando esté terminado será donado a la libreria los ficheros fuente y objeto en su forma definitiva y ejecutable). Por poner un ejemplo, el algoritmo usado en la división podria haberse hecho de otra manera.

Si solamente hubiesemos hecho un bucle que restará al dividendo (64 bit) el divisor (32 bit) hasta que nos diese un resto menor que el divisor, habriamos tenido que hacer 4294967295 restas de 64 bit en el peor de los casos.

Si solamente hubiesemos hecho 2 bucle que restarán al dividendo en segmentos de 48 bit el divisor (la misma técnica que nos enseñaron en el colegio), solamente habriamos tenido que hacer 65535 * 2 restas en el peor de los casos.

Si solamente hubiesemos hecho 4 bucles que restarán al dividendo en segmentos de 40 bit el divisor, solamente habriamos tenido que hacer 255 * 4 restas en el peor de los casos. Por desgracia he usado esta opción sin pensar en las siguientes.

Si solamente hubiesemos hecho 8 bucles que restarán al dividendo en segmentos de 36 bit el divisor, solamente habriamos tenido que hacer 15 * 8 restas en el peor de los casos.

Si solamente hubiesemos hecho un bucle que se repitiera 32 veces que restará al dividendo en segmentos de 33 bit el divisor, solamente habriamos tenido que hacer 32 restas en el peor de los casos. Este último es el más complicado y más largo, pero es el más rápido. Es fácil hacerlo usando las palabras de rotación bit Izda/derecha. Estooo... como cuando decidí probar la primera versión no corria, en vez de averiguar donde cometí el error, me decidí por escribir la versión más rápida de todas. ¿ Habeis adivinado Cuál ?.

Tengo que volver a recordar que todo esto es un experimento hecho en solamente 4 dias y en ratos libres, pero advierto que todas las pruebas que he hecho han dado el resultado correcto.

Si alguien advierte alguna burrada o resultado incorrecto, por favor comuniquemelo cuanto antes.

Hay gente que se preguntan si alguna vez existirá un sistema Forth de 32 bit. Bueno, pues según la poca información que yo tengo, existió en 1986 un sistema Forth de 32 bit de direcciones y stack para el ATARI ST, Forthmacs (Bradley Forthware). Los primeros articulos para un Forth 32 bit ("Compatible Forth on a 32-bit machine", "Extended addressing Wordset") aparecieron en 1984.

Actualmente han ganado la batalla las máquinas de 16-bit (lease IBM PC y compatibles), pero la única gran diferencia es que el direccionamiento en una máquina de 16-bit es por segmentos de 64 K (a base de punteros de 32 bit) y en una máquina de 32 bit el direccionamiento es lineal. Tal es la influencia de los PCs en el mercado que las nuevas máquinas de 32 bit por compatibilidad están obligadas a emular el direccionamiento por segmentos de 64 K del PC por compatibilidad.

Salvador Merino
Fuengirola (Málaga).
